

# PATENT ABSTRACTS OF JAPAN

701

(11)Publication number : 11-249910

(43)Date of publication of application : 17.09.1999

(51)Int.Cl.

G06F 9/46  
G06F 9/46  
G06F 3/06

(21)Application number : 10-046741

(71)Applicant : HITACHI LTD

(22)Date of filing : 27.02.1998

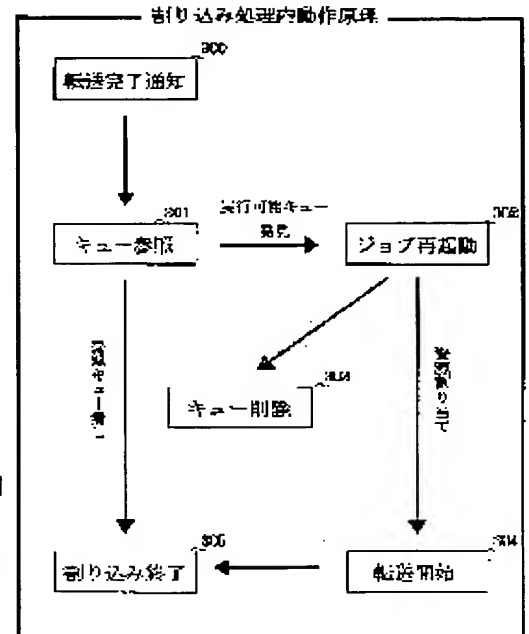
(72)Inventor : KOBAYASHI NAOTAKA  
NAGASE NORIKAZU  
OGASAWARA YUTAKA  
KUWABARA HIROSHI

## (54) STORAGE SUBSYSTEM AND DATA TRANSFER CONTROL METHOD

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To attain the effective use of a processor by reactivating only a job that is registered in a queue while referring to a queue table and starting the transfer of data when the jobs including the reactivation through the allocation of resources are executed.

**SOLUTION:** The reference is made to a queue table which is used for the queue registering (301). In this step 301, the registered queues are searched and an executable queue is detected. The detected executable queue is equal to a job queue that is registered after the current job queue which received the end notification and also equal to a queue that is logically connected to its preceding queue on the queue table. The transfer job of the queue is set in a transfer end wait state when the queue is registered and only this job is reactivated (302). Then the activated job is immediately deleted (303). Then the transfer resources are allocated to the activated job and this job is instructed to start the transfer of data (304). The interrupt processing is ended after the transfer of data is started (305).



## LEGAL STATUS

[Date of request for examination]

05.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-249910

(43) 公開日 平成11年(1999) 9月17日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/46

識別記号

3 4 0

F I

G 0 6 F 9/46

3 4 0 A

C

3/06

3 0 1

3/06

3 0 1 M

審査請求 未請求 請求項の数 2 O L (全 11 頁)

(21) 出願番号 特願平10-46741

(22) 出願日 平成10年(1998) 2月27日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72) 発明者 小林 直孝

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(72) 発明者 長瀬 則和

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(72) 発明者 小笠原 裕

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

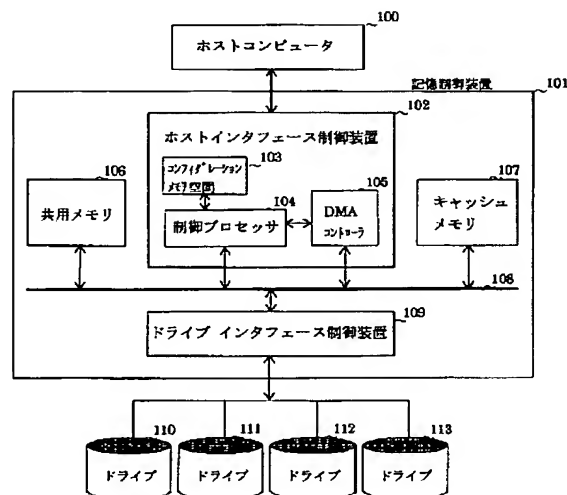
(54) 【発明の名称】 記憶サブシステム及びデータ転送制御方法

(57) 【要約】

【課題】複数のホストコンピュータと記憶装置（ストレージ）間でチャネルを介したデータ転送制御に関し、頻繁なデータ入出力要求からなる多重転送ジョブを、効率良く管理し、円滑な資源展開を可能とし、プロセッサの付加を軽減する。

【解決手段】ジョブ起動時に転送資源が空いているかどうかをチェックし、空いていれば資源を割り当てて転送を開始する。もし、空いていない場合は転送資源待ちにする代わりに転送完了待ちとしてキューテーブルに次回転送の予約をする。転送完了待ちとすることでジョブでは再起動を行う必要はない。そして、再起動～資源割り当ての作業を現在転送中のジョブが割り込みの完了通知を行うタイミングで行うものとする。この時、キューテーブルを参照してキュー登録されている1ジョブのみの再起動を行い転送を開始する。素早く目的のジョブが発見できるように、キュー構造にはNEXTキューのポインタ情報をもたせる。

図 1



## 【特許請求の範囲】

【請求項1】データを格納する複数の記憶装置と、前記記憶装置とホストコンピュータとのデータ転送を制御する記憶制御装置とを有する記憶サブシステムにおいて、前記記憶制御装置は、データ入出力要求があったときはジョブを起動し、データ転送が実行されないジョブを管理し、管理するジョブを連続して実行するよう制御する制御部と、

前記ジョブと連続して実行される前記ジョブの順序とを登録する手段とを有することを特徴とする記憶サブシステム。

【請求項2】データを格納する複数の記憶装置とホストコンピュータとのデータ転送を制御する記憶制御装置のデータ転送制御装置であって、データ入出力要求があったときはジョブを起動し、データ転送が実行されないジョブはキュー登録し、実行中のジョブが終了した場合はキュー登録されたジョブを起動することを特徴とするデータ転送制御装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、記憶装置（ストレージ）を有する記憶サブシステムに関し、特にホストコンピュータと記憶装置（ストレージ）間での記憶制御装置を介したデータ転送の際に多重転送ジョブを効率良く実行する技術に関する。

## 【0002】

【従来の技術】従来の技術では、ジョブは〈1〉入力データの準備〈2〉起動〈3〉実行待ちの制御〈4〉資源の割り当て〈5〉実行〈6〉資源の解放〈7〉出力

〈8〉終了までを一連の流れとして制御し、これが一般的なオペレーティングシステムによるジョブ管理方式である。転送ジョブにおいても同様であり、複数の転送ジョブが重なると、実行されるジョブは1つだけなので残りのジョブは転送資源解放待ちの待ち状態となり、転送資源解放時に再び〈2〉～〈5〉の処理が繰り返される。つまりジョブの多重度が増すと優先順位の低いジョブはなかなか資源が割り当てられずに、何度も状態遷移を繰り返すためプロセッサを不要に使用してしまう。また転送終了から次の転送起動までの間に優先順位の低いジョブを起動するのは時間のロスである。

【0003】これと類似のプロセス（タスク）の管理において、同様の問題を解決するために情報処理システム内で動作するプロセス（タスク）を一括して管理し、システムの処理能力の向上及びシステムの機能拡大に対応可能とするプロセス管理方式が特開平7-175671号に開示されている。

## 【0004】

【発明が解決しようとする課題】特開平7-175671号に開示された技術は、プロセス（タスク）の管理であるので、ジョブにおける〈5〉実行に相当する部分に限定さ

れていた。また、この技術は一旦停止したプロセスを再起動する際に、他のプロセス（タスク）と識別するために任意の識別情報を使用しすべてのプロセス（タスク）を管理することで不要なプロセスの起動を抑止している。本発明ではジョブ管理における不要な状態遷移を抑止するが、転送ジョブが重なった場合の転送資源開放時においてジョブの中で再起動を必要としない方式及び記憶サブシステムを提供する。

## 【0005】

【課題を解決するための手段】前述した課題を実現するために、本発明は、記憶サブシステムにおいて、ジョブ多重の増加に伴い効果を発揮するジョブ管理、転送起動方式を提供する。

【0006】データ転送を行うとき、ジョブ多重が発生すると1つのジョブ以外は全て転送資源待ち（待ち状態）となってしまう、転送終了時に次回転送起動ジョブ（最優先のジョブ）を待ち状態のジョブの中から見つけ出す必要がある。

【0007】従来の技術では、ここで一斉にジョブを再起動していた。従って、ジョブの多重度が増せば増す程効率が悪くなり、時間も掛かってしまったが、本発明では、転送ジョブをキュー管理することでこれを解決することができる。本発明では、ジョブ起動時に転送資源が空いているかどうかをチェックし、空いていれば、当然資源を割り当てて転送を開始する。もし、空いていない場合は、転送資源待ちにする代わりに転送完了待ちとしてキューテーブルに次回転送の予約をする。転送完了待ちとすることでジョブでは、再起動を行う必要はない。しかし、実際は転送完了待ちどころか資源待ちであるため、現在転送中のジョブが割り込みの完了通知を行うタイミングで再起動～資源割り当ての作業を行うものとする。

【0008】再起動～資源割り当ての作業を行う時、キューテーブルを参照することでキュー登録（予約）されている1ジョブのみの再起動を行い転送を開始することができる。キューテーブル参照時に、素早く目的のジョブが発見できる様に、キュー構造にはNEXTキューのアドレス情報を持たせ、キュー登録したジョブは、常に一つ前に登録したジョブキューに連結させる。したがって、転送完了割り込み時にキューテーブルを参照した際には、NEXTキューポインタが示す転送ジョブを起動することで効率的な連続転送が可能となる。

## 【0009】

【発明の実施の形態】以下、本発明の実施例を図面により説明する。図1は、本発明の実現装置である記憶制御装置101とそれを繋いだホストコンピュータ100との構成図である。本装置はチャネル側の処理を行うホストインタフェース制御装置102、ドライブ側の処理を行うドライブインタフェース制御装置109及び、不揮発性機構を有するキャッシュメモリ107、共用メモリ

106で構成し、各制御装置から信号線を介してキャッシュメモリ107、共用メモリ106へのデータ転送、制御を可能としている。

【0010】記憶制御装置101では、まずホストコンピュータ100の入出力データ要求に対しホストインタフェース制御装置102内の制御プロセッサ104が要求に応じて下部に処理を展開する。入出力データ要求によるデータ転送は、ホストインタフェース制御装置102内のコンフィグレーションメモリ空間103にセットされるエントリ情報に基づいて実行する。

【0011】READコマンドに対しては、要求データをドライブインタフェース制御装置109を介してドライブ群(110~113)からキャッシュメモリ107へステージングし、キャッシュメモリ107にステージングされたデータはホストインタフェース制御装置102内の制御プロセッサ104から制御信号を受けたDMAコントローラ105によりキャッシュメモリ107からホストコンピュータ100へのデータ転送を行う。またWRITEコマンドに対しては、要求データ分のキャッシュスロットをキャッシュメモリ107内に確保し、ホストインタフェース制御装置102内の制御プロセッサ104から制御信号を受けたDMAコントローラ105がホストコンピュータ100からキャッシュ107へのデータ転送を行い、さらにドライブインタフェース制御装置109を介してキャッシュ107からドライブ群(110~113)へ書き込まれる。

【0012】図2、図3は本発明における動作原理の概要を説明する図である。以下図2、図3を用いて本発明の動作原理の概要を説明する。図2は、1転送ジョブの起動から終了までを示している。まず、ジョブ起動(200)において実行可能となったジョブは転送資源(図1、DMAコントローラ105)が開放されていれば、転送を開始(201)させる。もし、別ジョブの実行により転送資源(図1、DMAコントローラ105)が空いていなければ、キューテーブル(図4で詳細を説明)に転送キューとして転送ジョブのエントリ情報を登録(202)する。ジョブの状態は、この時点で転送を開始した場合もキュー登録した場合も転送完了待ち(203)の待ち状態とする。転送完了待ち(203)となったジョブは転送資源(図1、DMAコントローラ105)からの転送完了通知のプロセッサ割り込み後ジョブを終了させる(204)。

【0013】転送完了通知割り込みからの処理については図3を使用して説明する。転送完了通知(300)の延長の処理として割り込み制御プログラムで以下の処理を行う。まず、図2でキュー登録に使用したキューテーブルを参照(301)する。ここでは登録キューをサーチし、実行可能なキューを発見する(存在するなら)。実行可能なキューが存在するなら、完了通知を受けた現在のジョブキュー(この時点では削除されて存在してな

い)の次に登録されたジョブキューであり、キューテーブル上では論理的に前のジョブキューに繋がれたキューであることを意味する。このキューの転送ジョブは、キュー登録を行った時点で転送完了待ちの待ち状態となっており、この1ジョブのみを再起動(302)させる。起動したジョブキューは、直ちに削除する(303)。そして、起動したジョブに転送資源を割り当てて転送を開始(304)させる。転送開始後は、割り込み処理を終了させる(305)。一方、キューの参照(301)で実行可能なキューを発見できなかったなら、登録キュー無しにより同じく割り込み処理を終了させる(305)。

【0014】以上が図2、図3で示す本発明における動作原理の概要であるが、基本的に転送データの準備をジョブで完了していれば特にジョブで実行する必要はないと考え、本方式を実施している。割り込みの延長処理では未完了のデータがある限り、キューから情報を取り出し、転送実行を続けキューが空となった時点で処理を止める方式である。

20 【0015】図4は、データ転送キューテーブルの構造を示す図である。キュー管理にはテーブル1、テーブル2の2つのテーブルを使用する。実際のジョブの情報を登録するデータ転送キューはテーブル2の方であり、テーブル1はこのテーブル2を管理する上で使用する。

【0016】テーブル2のキュー構造は、当該キューの次に実行されるキューのアドレスを指すNEXTキューポインタ格納エリア(404)、データ入出力要求におけるI-T-L-X情報格納エリア(405)及び、転送ジョブのエントリ情報の格納アドレスを指す転送エントリポインタ格納エリア(406)からなる。また、テーブル1のキュー構造は、テーブル1における登録されたキューのうち先頭のキューアドレスを指す登録管理先頭ポインタ(400)、同じく登録されたキューのうち最終のキューアドレスを指す登録管理最終ポインタ(401)、またテーブル1における空きキューのうち先頭のキューアドレスを指す空き管理先頭ポインタ(402)、同じく空きキューのうち最終のキューアドレスを指す空き管理最終ポインタ(403)からなる。

40 【0017】図5から図9に、キュー操作一連の動作を示す。以下、各フローチャート図を用いてキュー操作の原理を説明する。図5、図6は、図4で示した2つのキューテーブルの初期化処理を示す。図5はテーブル1の初期化である。ステップ500〜ステップ503においてポインタ格納エリアに初期値をセットする。登録管理先頭ポインタ(400)及び、登録管理最終ポインタ(401)には自ポインタアドレスを初期値に使用し、テーブル2に一つもキューが登録されていない状態を示す。空き管理先頭ポインタ(402)及び、空き管理最終ポインタ(403)の初期値にはそれぞれテーブル2上の先頭キューアドレス、最終キューアドレスをセット

する。

【0018】図6はテーブル2の初期化である。ステップ600でテーブル2の先頭のキューをカレントキューとする。カレントキューが最終のキューを超えるまでステップ601の判定でステップ602へ移行し、カレントキューのテーブルアドレスを取得する。ステップ603でカレントキューが最終キューならカレントキューのNEXTキューポインタに最終キューアドレスを設定する

(ステップ604)。ステップ603でカレントキューが最終キューでないならNEXTキューポインタにはテーブル上の次のキューアドレスをセットする(ステップ605)。この処理によって、カレントキューと次のキューが連結され、最終的にはすべてのキューがポインタで繋がれた形となる。次いで、ステップ606でカレントキューエリアを初期化(ゼロクリア)する。ステップ607で次のキューをカレントキューにセットしステップ601に戻る。ステップ601でカレントキューが最終キューを超えた時、初期化処理を終了する。

【0019】図7は、キュー登録処理の動作を示す。キューの登録は、テーブル1(図4)における空き管理先頭ポインタ(402)が指すアドレス上のキューを登録対象のキュー(カレントキュー)として転送情報を登録するものである。まず、ステップ700で登録キューの有り無しを判定する。もしすでに登録されたキューがあるならステップ702へ、もし登録されたキューが一つもないなら701へ移行し、フラグで識別する。ステップ703で空きキューの有り無しを判定する。キューが一つも空いていないならステップ708へ移行し、登録処理を終了する。キューが空いているならステップ704へ移行し、テーブル1(図4)から空き管理先頭ポインタ(402)を取得し、このポインタが指すアドレスのキューをカレントキューにする(ステップ705)。

【0020】次いで、登録管理最終ポインタ(401)、カレントのNEXTキューポインタを取得する(ステップ706、ステップ707)。カレントキューが最後の空きキューなら(ステップ709)、空き管理先頭ポインタ(402)及び登録管理最終ポインタ(401)に自ポインタアドレスをセットし(ステップ711、712)、カレントキューが最後のキューでないなら、新たな空き管理先頭ポインタ(402)としてカレントキューのNEXTキューポインタが示すキューアドレスをセット(ステップ710)し、カレントキューには登録情報をセットする(ステップ713)。カレントのNEXTキューポインタには繋ぐ登録キューが存在しないので自ポインタアドレスをセットする(ステップ714)。

【0021】次に、ステップ700の登録キューの有り無し判定の結果、フラグがたっているなら(ステップ715)、カレントキューを一つ前に登録したキューに繋ぐ必要があるため(ステップ716)、一つ前のキューのNEXTキューポインタにカレントキューのテーブルアド

レスをセットする。もし、ステップ700の判定でフラグがたっていないなら(ステップ715)、カレントキューが登録キューの先頭であることを意味するので登録管理先頭ポインタ(400)にカレントのキューアドレスをセットする(ステップ717)。キューアドレスをセットする(ステップ715)。最後に、新たな登録管理最終ポインタ(401)としてカレントのキューアドレスをセットする。

【0022】図8は、キュー検索処理を示す。キューの検索では、テーブル1(図4)における登録管理先頭ポインタ(400)が指すアドレス上のキューを実行対象のキュー(カレントキュー)として検索するものである。ステップ800で登録キュー有り無し判定をする。キューが登録されていないければ、ステップ805に移行し検索処理を終了する。キューが登録されているなら、テーブル2(図4)から登録管理先頭ポインタ(400)、登録管理最終ポインタ(401)を取得する(ステップ801、ステップ802)。

【0023】次いで、最後に登録されたキューのNEXTキューポインタを取得し(ステップ803)、ステップ801で取得した登録管理先頭ポインタをカレントキューにセットする(ステップ804)。このカレントのキューポインタが最後に登録されたキューのNEXTキューポインタと一致しない間(ステップ806)、カレントキューが実行可能キューであるか判定する(ステップ807)。ここでの実行可否は、SCSIにおけるCA状態(Contingent allegiance)のチェックである、カレントキューのI-T-L-Xから当該ジョブがCA状態でないかを確認する。カレントキューが実行可能であるなら、実行可能キュー通知を行う(ステップ811)。本通知を受けた割り込み処理により、カレントキューのジョブが再起動され、転送が始まる。実行可能キュー通知後は、直ちにカレントキューを削除し(ステップ812)、検索処理を終了する。

【0024】一方、ステップ807の判定でカレントキューが実行可能キューでないと判断した場合は、カレントキューポインタを退避し(ステップ809)、次の登録キューをカレントキューにセットし、ステップ806の判定に戻る。この判定で、カレントのキューポインタが最後に登録されたキューのNEXTキューポインタと一致した時点で検索処理を終了する。

【0025】図9は、キュー削除処理を示す。キューの削除は、キューの検索と一連の処理である。

【0026】実行可能な登録キューを発見したなら、そのキューが削除処理のカレントキューとなる。

【0027】まず、テーブル1から登録管理先頭ポインタ(400)、登録最終ポインタ(401)を取得し、最後に登録されたキューのNEXTキューポインタを取得する(ステップ900、ステップ901、ステップ902)。同じくテーブル1から空き管理先頭ポインタ(4

10

20

30

40

50

02)、空き管理最終ポインタ(403)を取得する(ステップ903、ステップ904)。カレントキューが登録キューの先頭、または最終なら(ステップ905)、ステップ906へ移行し、もし登録されているキューが一つしかないなら空き管理先頭ポインタ(402)及び空きキュー終端ポインタ(403)に自ポインタアドレスをセットし(ステップ912、ステップ913)、カレントキュー以外にもキューが登録されていれば、ステップ907の判定へ移行する。

【0028】カレントキューが登録されている先頭のキューなら、登録管理先頭ポインタ(400)にカレントのNEXTキューポインタが示すキューアドレスをセットし(ステップ911)、ステップ907で先頭のキューでないなら、登録管理最終ポインタ(401)にひとつ前に登録した(図8のステップ809で退避した)キューアドレスをセットし、このキューのNEXTキューポインタにカレントキューのNEXTキューポインタが示すキューアドレスをセットする(ステップ910)。ステップ905の判定でカレントキューが先頭または最終でないならステップ910と同様にひとつ前に登録したキューのNEXTキューポインタにカレントキューのNEXTキューポインタが示すキューアドレスをセットする(ステップ908)。次にステップ914の判定で空きキューが無いなら、空き管理先頭ポインタ(402)にカレントのキューアドレスをセットし(ステップ916)、空きキューがあるなら、最終空きキューのNEXTポインタにカレントのキューアドレスをセット(ステップ915)し、ステップ917に移行する。次いで空き管理最終ポインタに(402)カレントのキューアドレスをセットし(ステップ917)、カレントキューは初期化(空きキューに)する。

【0029】以上がキュー操作の原理であるが、テーブル2の登録キュー及び空きキューのNEXTキューポインタは、必ずしもテーブル上の次のキューアドレスを指し示しているわけではない。初期状態でのみ、すべて空きキューとしてそれぞれのキューのNEXTキューポインタは先頭から最終まで順番に次のキューを指している。しかし、ジョブが起動されて削除されたキューは、最終の空きキューの後に繋がれ、自キューが最終の空きキューとなり、また、キュー検索で実行されなかったカレントキューは実行可能になるまで登録キューのままであるため、このようにキュー登録、キュー検索、キュー削除が頻繁に行われるうちに空きキュー、登録キューのテーブル上の並びはばらばらであり、登録可能な順番もしくは登録された順番ではない。各キューのNEXTキューポインタは、テーブル上不規則に並んだキューの中で自キューの次に登録可能なキュー、もしくは自キューの次に実行可能なキューを正確に示しており、テーブル1(図4)がその先頭と最終を管理することですべてのキュー操作

を円滑に行えるしくみとなっている。

【0030】

【発明の効果】以上の実施例に示したように、本発明によれば1ジョブの転送完了時には、次回転送のジョブがキューを参照することにより、即時に明確になるため不要なジョブを起動することなく、目的のジョブのみの起動ができる。これにより、転送と転送の継ぎ目に要するオーバーヘッドを最小限にすることができ、プロセッサを効率的に使用できる。これは、ジョブの多重度が増せば増すほど効果を発揮する。

【図面の簡単な説明】

【図1】本発明の実施例における記憶サブシステムの構成を示す図である。

【図2】本発明によるジョブの動作原理を説明するための図である。

【図3】本発明による割り込み処理の動作原理を説明するための図である。

【図4】本発明によるデータ転送キュー管理テーブルの構成を示す図である。

【図5】本実施例におけるキュー操作(キューテーブルの初期化)のフローチャートである。

【図6】本実施例におけるキュー操作(キューテーブルの初期化)のフローチャートである。

【図7】本実施例におけるキュー操作(キューの登録)のフローチャートである。

【図8】本実施例におけるキュー操作(キューの検索)のフローチャートである。

【図9】本実施例におけるキュー操作(キューの削除)のフローチャートである。

【符号の説明】

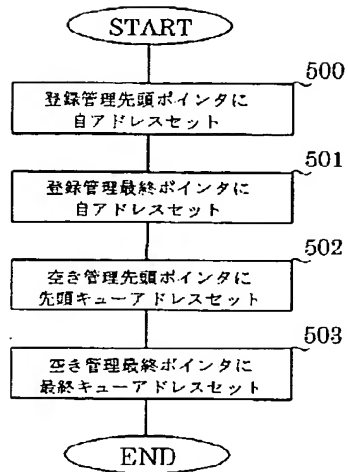
100	ホストコンピュータ	101	記憶制御装置
102	ホストインタフェース制御装置		
103	コンフィグレーションメモリ空間		
104	制御プロセッサ	105	DMAコントローラ
106	共用メモリ	107	キャッシュメモリ
108	信号線	109	ドライブ
110~113	ドライブ		
400	登録管理先頭ポインタ	401	登録管理最終ポインタ
402	空き管理先頭ポインタ	403	空き管理最終ポインタ
404	NEXTキューポインタ	405	I-T-L-X
406	転送エントリポインタ		





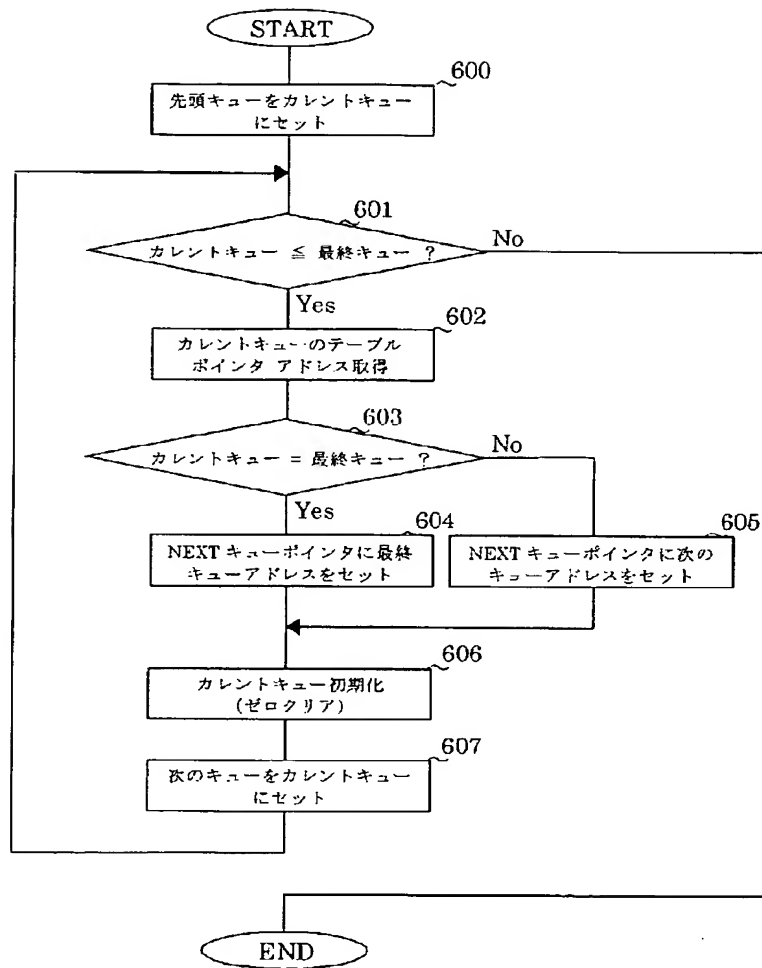
【図5】

図5

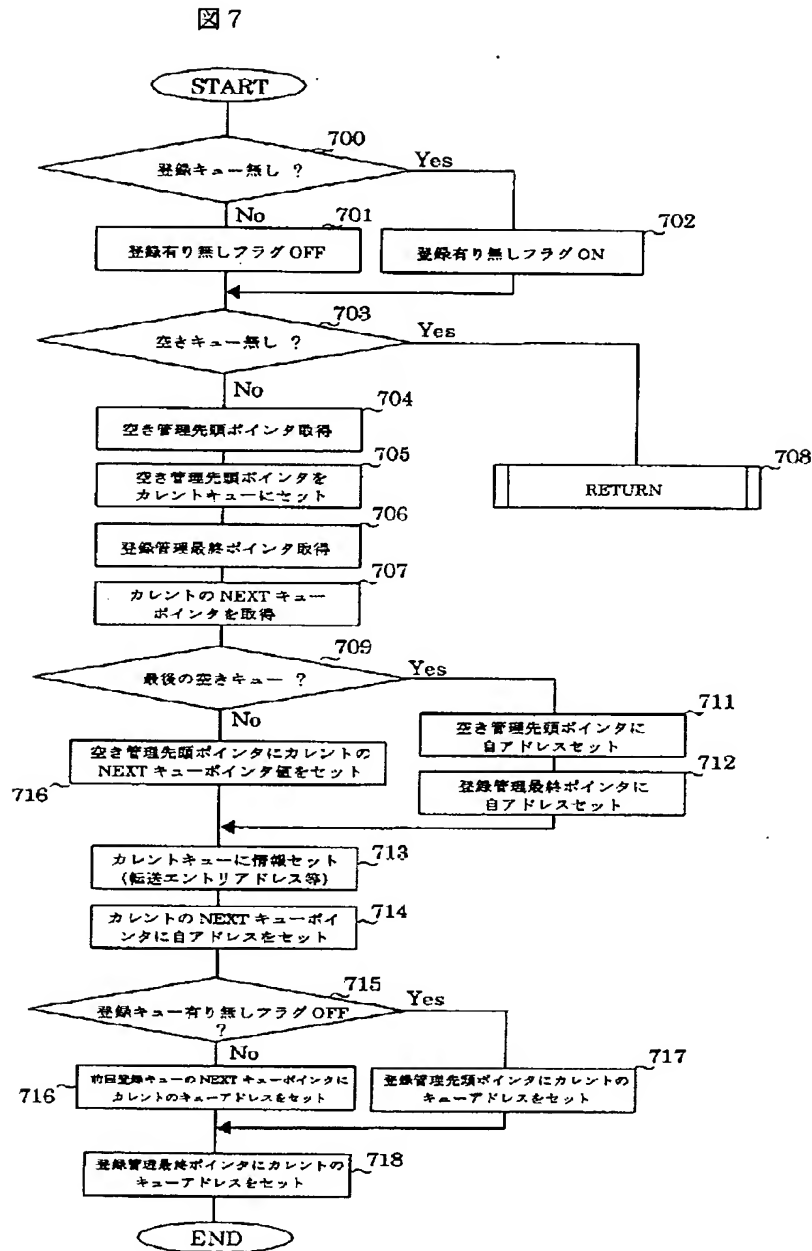


【図6】

図6

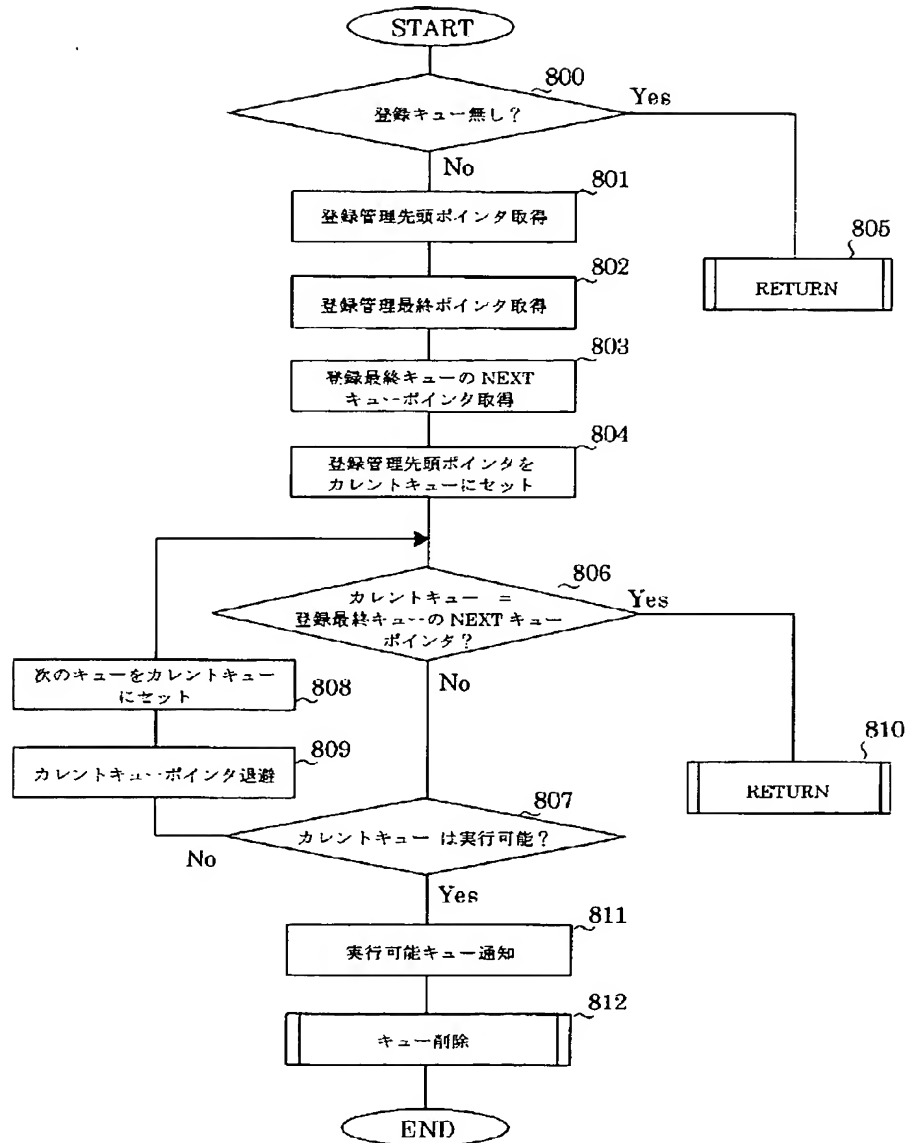


【図7】



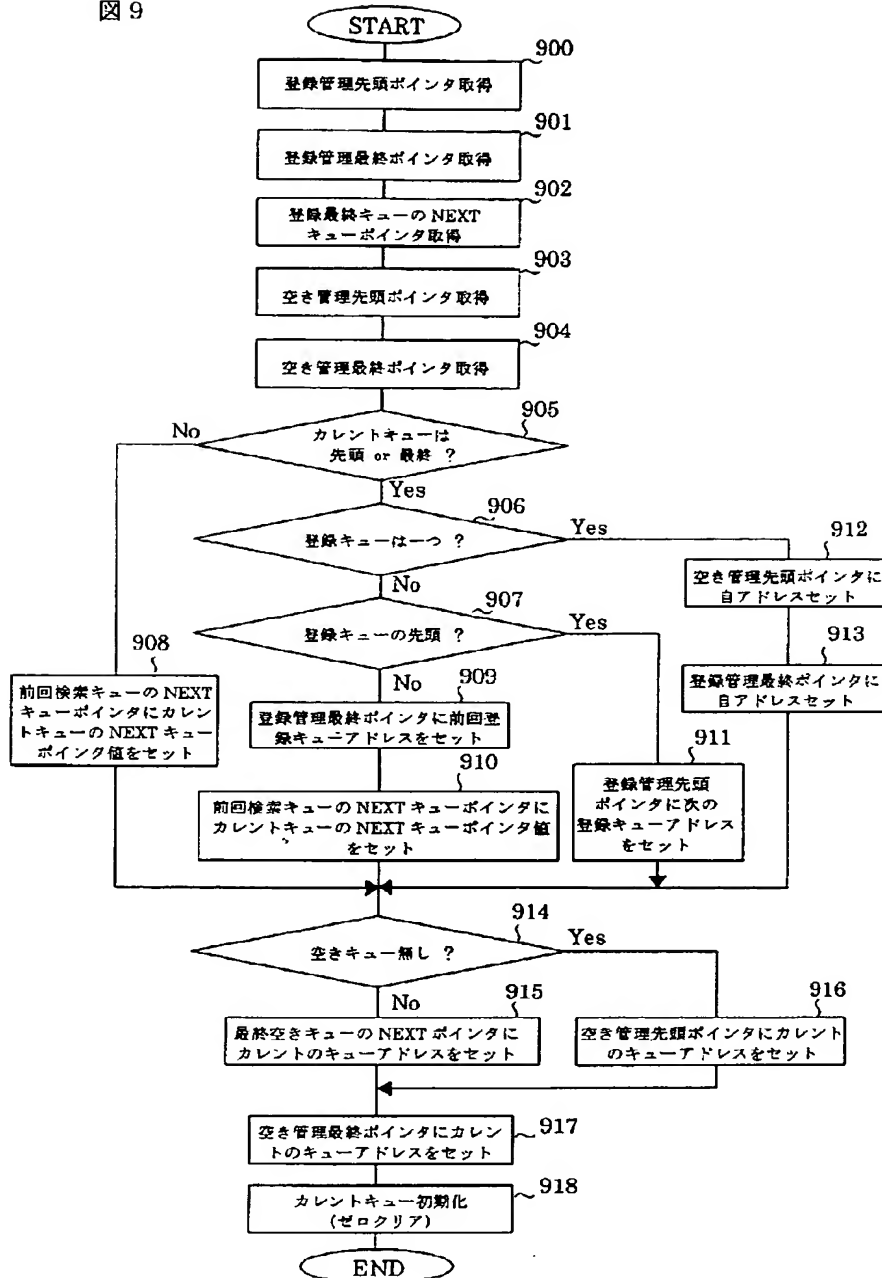
〔図 8〕

図 8



【図9】

図 9



フロントページの続き

(72)発明者 桑原 宏

神奈川県小田原市国府津2880番地 株式会  
社日立製作所ストレージシステム事業部内